



ITS
Institut
Teknologi
Sepuluh Nopember

KERJA PRAKTIK –

Pembuatan Aplikasi Perekaman Dokumen

PT. Taspen (Persero)

**DIREKTORAT PENDIDIKAN INSTITUT TEKNOLOGI
SEPULUH NOPEMBER SURABAYA**

Periode: 1 Agustus 2020 – 31 Agustus 2020

Oleh:

Raja Permata Boy 05111740000070

Edgar Andrew Hutauruk 05111740000166

Pembimbing Jurusan

Bilqis Amaliah S.Kom.,M.Kom.

Pembimbing Lapangan

Manaksak Siburian S.Si

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Informasi dan Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2020



KERJA PRAKTIK
Pembuatan Aplikasi Perekaman Dokumen
(Recording Dokumen)

PT. Taspen (Persero)

Periode: 1 Agustus 2020 – 31 Agustus 2020

Oleh:

Raja Permata Boy 05111740000070

Edgar Andrew Hutaaruk 05111740000166

Pembimbing Jurusan
Bilqis Amaliah S.Kom.,M.Kom.

Pembimbing Lapangan
Manaksak Siburian, S.Si.

DEPARTEMEN INFORMATIKA
Fakultas Teknologi Informasi dan Komunikasi
Institut Teknologi Sepuluh Nopember
Surabaya 2020

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

KERJA PRAKTIK

Pembuatan Aplikasi Perekaman Dokumen

Oleh:

Raja Permata Boy M S

05111740000070

Edgar Andrew H

05111740000166

Mengetahui,

Menyetujui,

Dosen Pembimbing

Pembimbing Kerja Praktik

Kerja Praktik



Manaksak Siburian S.Si



Bilqis Amaliah, S.Kom, M. Kom

Manaksak Siburian S.Si

Bilqis Amaliah, S.Kom, M. Kom

NIP. 197509142001122002

SURABAYA

November, 2020

[Halaman ini sengaja dikosongkan]

Pembuatan Aplikasi Perekaman Dokumen Taspen

Nama Mahasiswa : Raja Permata Boy
NRP : 05111740000070
Nama Mahasiswa : Edgar Andrew Hutauruk
NRP : 05111740000166
Departemen : Informatika FTEIC-ITS
Pembimbing Jurusan : Bilqis Amaliah, S.Kom., M.Kom.
Pembimbing Lapangan : Manaksak Siburian, S.Si

ABSTRAK

PT TASPEN (Persero) adalah salah satu perusahaan BUMN besar yang bergerak di bidang asuransi tabungan hari tua dan dana pensiun Pegawai Negeri Sipil (PNS). PT TASPEN sendiri terdiri dari beberapa departemen, salah satunya adalah divisi teknologi informasi.

Dalam departemen tersebut, mereka masih menyimpan database mereka di pusat data (belum ada aplikasi khusus) Sehingga kami sebagai mahasiswa magang diminta untuk membantu menyelesaikan problematika yang ada.

Untuk Menjawab hal tersebut dibutuhkan sebuah aplikasi perekaman dokumen berbasis web. Di sini, karyawan dapat menyimpan dokumen, mengedit dokumen dan mendownload dokumen yang sudah disimpan. Kami membuat aplikasi tersebut dengan menggunakan framework Laravel yang dipadukan dengan PHP, JavaScript dan tentunya Bootstrap. Untuk databasenya sendiri kami menggunakan database dari MySQL dan menggunakan package Spatie untuk mengatur role dari user.

Kata kunci: Laravel, PHP, JavaScript, Bootstrap, MySQL, Spatie

KATA PENGANTAR

Puji syukur kami haturkan kepada Allah SWT karena berkat rahmat-Nya kami dapat melaksanakan salah satu kewajiban kami sebagai mahasiswa Departemen Informatika, yakni Kerja Praktik (KP).

Kami menyadari masih ada kekurangan baik dalam pelaksanaan kerja praktik maupun penyusunan buku laporan ini. Namun, kami berharap buku laporan ini dapat menambah wawasan pembaca dan dapat menjadi sumber referensi. Kami mengharapkan kritik dan saran yang membangun untuk kesempurnaan buku laporan kerja praktik ini.

Melalui buku ini, kami juga ingin menyampaikan rasa terima kasih kepada orang-orang yang telah membantu, baik langsung maupun tidak langsung, dalam pelaksanaan kerja praktik hingga penyusunan laporan. Orang-orang tersebut antara lain adalah:

1. Kedua orang tua penulis.
2. Ibu Bilqis Amaliah S.Kom., M.kom. selaku dosen pembimbing Kerja Praktik.
3. Bapak Ary Mazharuddin Shiddiqi S.Kom., M.Comp., Ph.D, selaku koordinator Kerja Praktik.
4. Bapak Manaksak Siburian, S.Si. selaku pembimbing lapangan Kerja Praktik.

Surabaya, November 2020

Raja Permata Boy, Edgar Andrew Hutaaruk

DAFTAR ISI

LEMBAR PENGESAHAN	5
ABSTRAK	8
KATA PENGANTAR	9
DAFTAR ISI	10
BAB I	
PENDAHULUAN	15
1.1. Latar Belakang	15
1.2. Tujuan	15
1.3. Manfaat	15
1.4. Rumusan Masalah	16
1.5. Lokasi dan Waktu Kerja Praktik	16
1.6. Metodologi Kerja Praktik	16
1.7. Sistematika Laporan	18
 BAB II	
PROFIL PERUSAHAAN	21
2.1. Profil PT. Taspen (Persero)	21
2.2. Logo Perusahaan	22
2.3. Visi Misi Perusahaan	23
2.4. Struktur Organisasi	23

BAB III	
TINJAUAN PUSTAKA	25
3.1. Laravel	25
3.2. PHP	25
3.3. Bootstrap	26
3.4. Javascript	26
3.5. Spatie	26
3.6. SQL	27

BAB IV	
IMPLEMENTASI SISTEM	29
4.1. Home	29
4.2. Halaman Login	30
4.3. Halaman Register	31
4.4. Halaman Rekam Dokumen	32
4.5. Halaman Pencarian Dokumen	34
4.6. Halaman Pencarian Dokumen (Admin)	36
4.7. Home Controller	37
4.8. Data Controller	38
4.9. Model Dokumen	41
4.10. Model Karyawan	42
4.11. Model Klasifikasi	43
4.12. Model TipeKlasifikasi	44
4.13. Model UnitKerja	45
4.14. Model User	46
4.15. Migrasi Tabel User	47
4.16. Migrasi Tabel User Permission	48
4.17. Migrasi Tabel Unit Kerja	49

4.18. Migrasi Tabel Klasifikasi	50
4.19. Migrasi Tabel Tipe Klasifikasi	51
4.20. Migrasi Tabel Karyawan	52
4.21. Migrasi Tabel Dokumen	53
 BAB V	
PENGUJIAN DAN EVALUASI	55
5.1. Tujuan Pengujian	55
5.2. Kriteria Pengujian	55
5.3. Skenario Pengujian	55
5.4. Evaluasi Pengujian	59
 BAB VI	
KESIMPULAN DAN SARAN	61
6.1. Kesimpulan	61
6.2. Saran	61
 DAFTAR PUSTAKA	62
 BIODATA PENULIS I	63
 BIODATA PENULIS II	65

DAFTAR GAMBAR

Gambar 2.1 Lambang dan Logo PT. TASPEN

Gambar 2.2 Struktur Organisasi PT. TASPEN

Gambar 4.1 Halaman Home

Gambar 4.2 Halaman Login

Gambar 4.3 Halaman Register

Gambar 4.4 Halaman Rekam Dokumen

Gambar 4.5 Halaman Pencarian Dokumen

Gambar 4.6 Halaman Pencarian Dokumen (admin)

Gambar 5.3.1. Pengujian Input Dokumen

Gambar 5.3.2. Pengujian Filter Dokumen dengan Nama Dokumen

Gambar 5.3.2. Pengujian Filter Dokumen dengan Tanggal
Dokumen

Gambar 5.3.3. Pengujian Menghapus Dokumen melalui ID

Gambar 5.3.4. Pengujian MengeditDokumen melalui ID

DAFTAR TABEL

Tabel 5.4.. Tabel Evaluasi Pengujian Kebutuhan

DAFTAR KODE

Kode 4.1. Halaman Home

Kode 4.2. Halaman Login

Kode 4.3.1. Halaman Register

Kode 4.3.2. Halaman Register - 2

Kode 4.4.1. Halaman Rekaman Dokumen

Kode 4.4.2. Halaman Rekaman Dokumen - 2

Kode 4.4.3. Halaman Rekaman Dokumen - 3

Kode 4.5.1. Halaman Pencarian Dokumen

Kode 4.5.2. Halaman Pencarian Dokumen - 2

Kode 4.6. Kode Authorization Laravel

Kode 4.7. Kode Home Controller

Kode 4.8.1. Kode Data Controller

Kode 4.8.2. Kode Data Controller - 2
Kode 4.8.3. Kode Data Controller - 3
Kode 4.8.4. Kode Data Controller - 4
Kode 4.8.5. Kode Data Controller - 5
Kode 4.8.6. Kode Data Controller - 6
Kode 4.9. Kode Model Dokumen
Kode 4.10. Kode Model Karyawan
Kode 4.11. Kode Model Klasifikasi
Kode 4.12. Kode Model TipeKlasifikasi
Kode 4.13. Kode Model Unit Kerja
Kode 4.14. Kode Model User
Kode 4.15. Kode Migrasi Tabel User
Kode 4.16.1. Kode Migrasi Tabel User Permission
Kode 4.16.2. Kode Migrasi Tabel User Permission - 2
Kode 4.16.3. Kode Migrasi Tabel User Permission - 3
Kode 4.17. Kode Migrasi Tabel Unit Kerja
Kode 4.18. Kode Migrasi Tabel Klasifikasi
Kode 4.19. Kode Migrasi Tipe Klasifikasi
Kode 4.20. Kode Migrasi Tabel Karyawan
Kode 4.21.1. Kode Migrasi Tabel Dokumen
Kode 4.21.2. Kode Fungsi Drop Tabel Dokumen jika exist

BAB I

PENDAHULUAN

1.1. Latar Belakang

PT TASPEN (Persero) adalah salah satu perusahaan BUMN besar yang bergerak di bidang asuransi tabungan hari tua dan dana pensiun Pegawai Negeri Sipil (PNS). PT TASPEN sendiri terdiri dari beberapa departemen, salah satunya adalah divisi teknologi informasi.

Dalam departemen tersebut, masih belum adanya aplikasi web sebagai tempat untuk menyimpan, mengedit dan menampilkan dokumen yang ada. Sehingga kami sebagai mahasiswa magang diminta untuk membantu menyelesaikan problematika yang ada. Kami berencana untuk membangun aplikasi berbasis web dengan menggunakan Laravel sebagai framework (kerangka kerja).

1.2. Tujuan

Tujuan kerja praktik ini adalah untuk menyelesaikan kewajiban kuliah kerja praktik di Institut Teknologi Sepuluh Nopember dengan beban dua SKS. Selain itu juga untuk membangun sistem perekaman dokumen yang ada di Taspen yang belum ada sebelumnya.

1.3. Manfaat

Dengan adanya sistem yang akan dibuat, diharapkan dapat membantu PT. Taspen (Persero) dalam merekam dan menyimpan dokumen yang ada.

1.4. Rumusan Masalah

Berikut ini rumusan masalah pada kerja praktik pembuatan Sistem Perekaman Dokumen:

1. Bagaimana cara membantu pegawai dalam menyimpan dokumen yang ada?
2. Bagaimana cara membantu pegawai dalam mengakses dokumen yang telah disimpan?

1.5. Lokasi dan Waktu Kerja Praktik

Kerja praktik ini dilaksanakan pada waktu dan tempat sebagai berikut:

Lokasi	: Online dan offline
Waktu	: 1 Agustus 2020 – 31 Agustus 2020
Hari Kerja	: Senin - Jumat
Jam Kerja	: 08.00 WIB – 15.00 WIB

1.6. Metodologi Kerja Praktik

Tahapan pengerjaan kerja praktik dapat dijabarkan sebagai berikut:

1.6.1. Perumusan Masalah

Dalam tahap ini kami perlu mengetahui permasalahan apa saja yang terjadi dan dapat diselesaikan atau dioptimasi. Lalu kami juga perlu mengetahui semua kebutuhan dalam permasalahan tersebut.

1.6.2. Studi Literatur

Setelah ditentukan rumusan masalah mengenai sistem yang akan dibuat, dilakukan studi literatur mengenai implementasinya. Pada tahap ini dilakukan proses pencarian, pembelajaran, dan pengumpulan informasi yang

berkaitan dengan implementasi sistem yang akan dibuat. Informasi dapat diperoleh dari internet ataupun dari proyek sebelumnya yang serupa dan memungkinkan untuk diimplementasikan.

1.6.3. Analisis dan Perancangan

Tahap ini meliputi penjelasan mengenai hasil dari studi literatur yang dilakukan. Dari beberapa metode yang ditemukan saat literasi dianalisa metode mana yang paling tepat dan efektif untuk digunakan untuk menyelesaikan permasalahan. Ditentukan bahasa pemrograman yang akan digunakan, batasan-batasan sistem yang dibuat, juga rancangan antarmuka aplikasi yang akan mudah digunakan pengguna.

1.6.4. Implementasi Sistem

Pada tahap ini dijelaskan implementasi program yang digunakan pada proses pembuatan sistem perekaman dokumen yang akan dibuat. Bagian ini meliputi penjelasan dari pembuatan sistem perekaman dokumen dengan menggunakan aplikasi. Sistem perekaman dokumen yang dikembangkan berbasis web, dengan framework laravel.

1.6.5. Pengujian dan Evaluasi

Pengujian sistem yang dilakukan merupakan pengujian terhadap fungsi-fungsi yang telah kami buat.

1.6.6. Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran dalam pengerjaan kerja praktik.

1.7. Sistematika Laporan

Laporan kerja praktik ini terdiri dari tujuh bab dengan rincian sebagai berikut:

1.7.1. Bab I Pendahuluan

Pada bab ini dijelaskan tentang latar belakang permasalahan, tujuan, waktu pelaksanaan, serta sistematika pengerjaan kerja praktik dan juga penulisan laporan kerja praktik.

1.7.2. Bab II Profil Perusahaan

Pada bab ini, dijelaskan secara rinci tentang profil PT. TASPEN (Persero) , tempat kami melaksanakan kerja praktik.

1.7.3. Bab III Tinjauan Pustaka

Pada bab ini, dijelaskan mengenai tinjauan pustaka dan literatur yang digunakan dalam penyelesaian kerja praktik di PT. TASPEN (Persero).

1.7.4. Bab IV Implementasi Sistem

Pada bab ini, berisi penjelasan tahap-tahap yang dilakukan untuk proses pembuatan Sistem Perekaman Dokumen.

1.7.5. Bab V Pengujian dan Evaluasi

Pada bab ini, dijelaskan tentang hasil pengujian dan evaluasi dari sistem yang telah dikembangkan selama pelaksanaan kerja praktik.

1.7.6. Bab VI Kesimpulan dan Saran

Pada bab ini, dipaparkan kesimpulan yang dapat diambil dan juga saran selama pengerjaan kerja praktik.

[Halaman ini sengaja dikosongkan]

BAB II

PROFIL PERUSAHAAN

2.1. Profil PT. Taspen (Persero)

PT Dana Tabungan dan Asuransi Pegawai Negeri (Persero) atau disingkat PT TASPEN (PERSERO) telah menjadi bagian yang tidak terpisahkan dari perjalanan panjang sejarah abdi Negara di Indonesia atas peran yang diberikan oleh Pemerintah Republik Indonesia dalam menyelenggarakan Program Asuransi Sosial Pegawai Negeri Sipil (PNS) yang terdiri dari Program Pensiun Pegawai Negeri Sipil (PNS) dan Tabungan Hari Tua (THT) dengan tujuan untuk meningkatkan kesejahteraan Pegawai Negeri pada saat memasuki usia pensiun.

Berawal dari Konferensi Kesejahteraan Pegawai Negeri yang diselenggarakan pada tanggal 25-26 Juli 1960 di Jakarta yang menghasilkan Keputusan Menteri Pertama RI Nomor 388/MP/1960 tanggal 25 Agustus 1960. Dalam Keputusan tersebut, Pemerintah menetapkan pentingnya pembentukan jaminan sosial sebagai bekal bagi Pegawai Negeri dan keluarganya di masa purna bakti. Kemudian pada tanggal 17 April 1963, Pemerintah mendirikan Perusahaan Negara Dana Tabungan dan Asuransi Pegawai Negeri (PN TASPEN) melalui Peraturan Pemerintah Nomor 15 Tahun 1963. Pembentukan Program Tabungan Hari Tua Pegawai Negeri ditetapkan dalam Peraturan Pemerintah No 9 tahun 1963 tentang Pembelanjaan Pegawai Negeri dan Peraturan Pemerintah Nomor 10 tahun 1963 tentang Tabungan Asuransi dan Pegawai negeri.

Dengan pengalaman yang sudah terbukti dalam memberikan pelayanan program asuransi sosial bagi ASN dan Pejabat Negara

dan lebih memberikan layanan yang terbaik bagi Peserta, TASPEN melakukan pembenahan dan mendekatkan diri kepada Peserta melalui 57 Kantor Cabang yang tersebar di Seluruh Indonesia terdiri dari 6 Kantor Cabang Utama, 7 Kantor Cabang tipe A, 14 Kantor Cabang tipe B, 19 Kantor Cabang tipe C dan 11 Kantor Cabang tipe D. Sebagai salah satu instansi pelayanan publik, TASPEN berkomitmen untuk senantiasa meningkatkan pelayanan bagi Aparatur Sipil Negara dan Pejabat Negara melalui berbagai macam inovasi, mulai dari digital-based service, layanan klim otomatis, layanan kunjungan nasabah hingga layanan klaim satu jam yang telah memperoleh sertifikasi ISO 9001: 2015. Hal ini merupakan komitmen Perseroan untuk terus meningkatkan kualitas layanan guna tercapainya kesejahteraan Aparatur Sipil Negara yang berkelanjutan.

2.2. Logo Perusahaan



Gambar 2.1 lambang dan logo PT TASPEN

2.3. Visi Misi Perusahaan

2.3.1. Visi

Menjadi Perusahaan Asuransi Sosial dan Dana Pensiun yang Unggul, Terpercaya dan Berkelanjutan demi mewujudkan kesejahteraan Peserta untuk meningkatkan nilai ekonomi dan sosial Indonesia.

2.3.1. Misi

Memastikan terwujudnya Layanan Terbaik dan Investasi yang Andal serta Kepemimpinan Inovasi Bisnis dan Transformasi Digital dengan didukung oleh Sumber Daya Manusia yang Amanah, Kompeten, Harmonis, Loyal, Adaptif dan Kolaboratif.

2.4. Struktur Organisasi



Gambar 2.2 Struktur Organisasi PT.Taspen (Persero)

[Halaman ini sengaja dikosongkan]

BAB III

TINJAUAN PUSTAKA

Pada bab ini, akan dijelaskan mengenai dasar teori yang digunakan selama proses kerja praktik.

3.1. Laravel

Laravel adalah kerangka aplikasi web dengan sintaks yang ekspresif dan elegan. Laravel mencoba menghilangkan kesulitan dalam pengembangan dengan mengurangi tugas-tugas umum yang digunakan di sebagian besar proyek web, seperti otentikasi, perutean, sesi, dan caching.

Laravel bertujuan untuk membuat proses pengembangan menyenangkan bagi pengembang tanpa mengorbankan fungsionalitas aplikasi.

Untuk tujuan ini, Laravel telah mencoba menggabungkan yang terbaik dari apa yang telah kami lihat di kerangka kerja web lain, termasuk kerangka kerja yang diimplementasikan dalam bahasa lain, seperti Ruby on Rails, ASP.NET MVC, dan Sinatra.

3.2. PHP

PHP Hypertext Preprocessor (PHP) adalah bahasa pemrograman yang memungkinkan pengembang web membuat konten dinamis yang berinteraksi dengan database. PHP pada dasarnya digunakan untuk mengembangkan aplikasi perangkat lunak berbasis web. Tutorial ini membantu Anda membangun basis dengan PHP.

PHP dimulai sebagai proyek open source kecil yang berkembang karena semakin banyak orang mengetahui betapa bermanfaatnya itu. Rasmus Lerdorf merilis versi pertama PHP pada tahun 1994.

PHP adalah KEHARUSAN bagi siswa dan profesional yang bekerja untuk menjadi Insinyur Perangkat Lunak yang hebat khususnya ketika mereka bekerja di Domain Pengembangan Web.

3.3. Bootstrap

Bootstrap is the world's most popular front-end open source toolkit, featuring Sass variables and mixins, responsive grid system, extensive pre-built components, and powerful JavaScript plugins.

3.4. Javascript

JavaScript (JS) adalah bahasa pemrograman terkompilasi ringan, ditafsirkan, atau hanya dalam waktu dengan fungsi kelas satu. Meskipun paling dikenal sebagai bahasa skrip untuk halaman Web, banyak lingkungan non-browser juga menggunakannya, seperti Node.js, Apache CouchDB, dan Adobe Acrobat.

JavaScript adalah bahasa berbasis prototipe, multi-paradigma, single-threaded, dinamis, mendukung gaya berorientasi objek, imperatif, dan deklaratif (misalnya pemrograman fungsional).

3.5. Spatie

Paket ini memungkinkan Anda untuk mengelola izin dan peran pengguna dalam database.

Setelah terinstal, Anda dapat melakukan hal-hal seperti ini:

```
// Menambahkan izin ke pengguna  
$ user-> givePermissionTo ('edit artikel');
```

```
// Menambahkan izin melalui peran  
$ user-> assignRole ('writer');
```

```
$ role-> givePermissionTo ('edit artikel');
```

3.6. SQL

MySQL Database Service adalah layanan database terkelola sepenuhnya yang memungkinkan organisasi menerapkan aplikasi cloud-native menggunakan database open source terpopuler di dunia. Ini 100% dikembangkan, dikelola, dan didukung oleh Tim MySQL.

SQL adalah singkatan dari Structured Query Language. SQL digunakan untuk berkomunikasi dengan database. Menurut ANSI (American National Standards Institute), ini adalah bahasa standar untuk sistem manajemen basis data relasional. Pernyataan SQL digunakan untuk melakukan tugas-tugas seperti memperbarui data pada database, atau mengambil data dari database.

[Halaman ini sengaja dikosongkan]

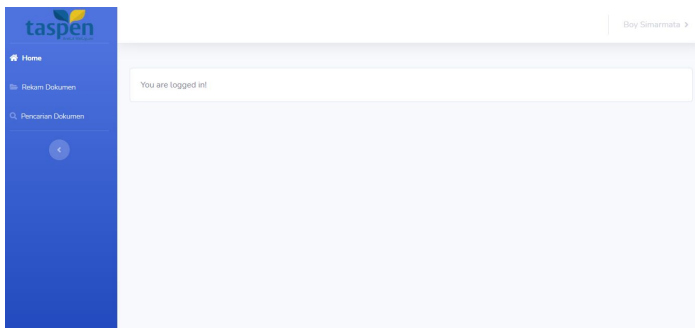
BAB IV

IMPLEMENTASI SISTEM

Pada bab ini menjelaskan tahap implementasi yang dilakukan pada aplikasi recording dokumen berbasis web pada PT. Taspen (Persero) selama kerja praktik.

4.1. Home

Home adalah halaman pertama yang diakses user ketika melakukan login pada web perekaman dokumen.



Gambar 4.1. Halaman Home

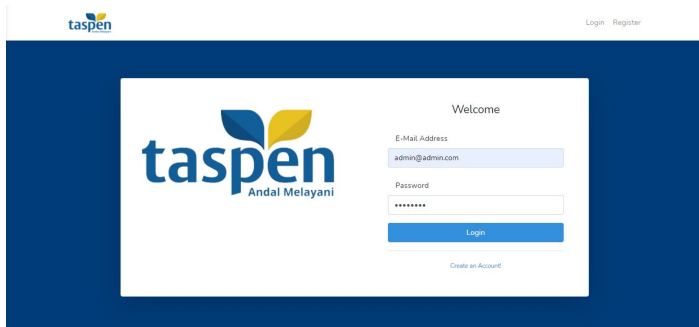
```
@extends('layouts.app')

@section('content')
<div class="container">
  <div class="row justify-content-center">
    <div class="col-md-8">
      <div class="card">
        <!-- @can('edit articles') -->
        <div class="card-header">{{ __('Dashboard' ) }}</div>
        <!-- @endcan -->
        <div class="card-body">
          @if (session('status'))
            <div class="alert alert-success" role="alert">
              {{ session('status') }}
            </div>
          @endif
          {{ __('You are logged in!') }}
        </div>
      </div>
    </div>
  </div>
</div>
@endsection
```

Kode 4.1. Halaman Home

4.2. Halaman Login

Pada halaman login, user yang sudah memiliki akun dapat menginputkan data email dan password yang telah didaftarkan sebelumnya.



Gambar 4.2 Halaman Login

```

<!-- layout: app -->
<div class="container">
  <div class="row justify-content-center">
    <div class="col-xl-10 col-lg-12 col-md-9">
      <div class="card shadow border-0">
        <div class="card-body p-4">
          <div class="row">
            <div class="col-lg-6 d-none d-lg-block bg-login-image"></div>
            <div class="col-lg-6">
              <div class="p-4">
                <div class="text-center">
                  <div class="text-gray-900 mb-4">Welcome</div>
                  <form method="POST" action="{ { route('login') }}">
                    <div class="form-group">
                      <div class="form-control">
                        <input type="text" class="form-control" value="{ { old('email') }}" required="" />
                      </div>
                      <div class="invalid-feedback"><div></div></div>
                    </div>
                    <div class="form-group">
                      <div class="form-control">
                        <input type="password" class="form-control" value="{ { old('password') }}" required="" />
                      </div>
                      <div class="invalid-feedback"><div></div></div>
                    </div>
                    <div class="text-center">
                      <button type="submit" class="btn btn-primary">Login</button>
                    </div>
                    <div class="text-center">
                      <a href="{ { route('register') }}">Create an Account</a>
                    </div>
                  </form>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

Kode 4.2. Halaman Login

ser y
eper
skri



r 4.3

```

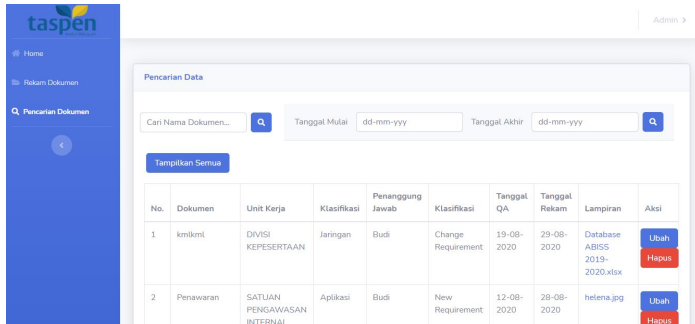
    "right">{{ __(
      'name') is-
    "right">{{ __(
      'email')
    "right">{{
      _error('pa

```

Hala

4.6. Halaman Pencarian Dokumen (Admin)

Pada halaman pencarian dokumen (admin), kurang lebih sama dengan halaman pencarian dokumen, tetapi jika user yang login sebagai admin dapat melakukan mengeditan rekaman dokumen dan penghapusan rekaman dokumen.



Gambar 4.6 Halaman Pencarian Dokumen (Admin)

```
@can('edit articles')  
<th>Aksi</th>  
@endcan
```

Kode 4.6. Kode Authorization Laravel

4.7. Home Controller

Controller ini berfungsi sebagai pengatur halaman home pada saat autentikasi user oleh web.

```
<?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class HomeController extends Controller
{
    /**
     * Create a new controller instance.
     *
     * @return void
     */
    public function __construct()
    {
        $this->middleware('auth');
    }

    /**
     * Show the application dashboard.
     *
     * @return \Illuminate\Contracts\Support\Renderable
     */
    public function index()
    {
        return view('home');
    }
}
```

Kode 4.7. Kode Home Controller

4.8. Data Controller

Controller ini berfungsi untuk mengatur jalannya data masuk dan keluar pada web.

```
<?php
namespace App\Http\Controllers;

use App\Document;
use App\ViewModel;
use App\Classifikasi;
use App\UnitKerja;
use Illuminate\Http\Response;

use Illuminate\Http\Request;
use League\CommonMark\Block\Element\Document;

class DataController extends Controller
{
    /**
     * Display a listing of the resource.
     *
     * @return \Illuminate\Http\Response
     */
    public function search(Request $request)
    {
        $batas = 10;

        $cari = $request->cari;

        $data = Document::where('NamaDokumen', 'like', "% " . $cari . "%")->orderBy('NamaDokumen', 'desc')->paginate($batas);
        $no = $batas * ($data->currentPage() - 1);

        return view('cari', ['data' => $data, ['no' => $no])->with('status', 'Ditemukan ' . count($data) . ' Dokumen');
    }

    public function filter(Request $request)
    {
        $request->validate(
            [
                'mulai' => 'required|before_or_equal:akhir',
                'akhir' => 'required',
            ]
        );

        $mulai = data('V.m.d', strtotime($request->mulai));
```

Kode 4.8.1. Kode Data Controller

```

$request->validate(
    [
        'mulai' => 'required|before_or_equal:akhir',
        'akhir' => 'required',
    ]
);

$mulai = date('Y-m-d', strtotime($request->mulai));
$akhir = date('Y-m-d', strtotime($request->akhir));

$batas = 10;

$data = Dokumen::whereDate('TanggalQA', '>=', $mulai)->whereDate('TanggalQA', '<=', $akhir)-orderBy('TanggalQA', 'desc')->paginate($batas);
$no = $batas * ($data->currentPage() - 1);

return view('cari', ['data' => $data], ['no' => $no])->with('status', 'Ditemukan ' . count($data) . ' Dokumen');
}

public function index()
{
    //
    $batas = 10;
    $karyawan = Karyawan::all();
    $datas = Dokumen::count();
    $data = Dokumen::orderBy('IdDokumen', 'desc')->paginate($batas);
    $no = $batas * ($data->currentPage() - 1);
    return view('cari', compact('data', 'no', 'karyawan'));
}

/**
 * Show the form for creating a new resource.
 *
 * @return \Illuminate\Http\Response
 */
public function create()
{
    //
    $karyawan = Karyawan::all();
    $klasifikasi = Klasifikasi::all();
    $tipe = TipeKlasifikasi::all();
    $unit = UnitKerja::all();
    return view('dokumen', compact('karyawan', 'klasifikasi', 'tipe', 'unit'));
}

/**

```

Kode 4.8.2. Kode Data Controller - 2

```

* @param \Illuminate\Http\Request $request
* @return \Illuminate\Http\Response
*/
public function store(Request $request)
{
    //
    // dd($request);
    $request->validate(
        [
            'NamaDokumen' => 'required',
            'id_kdunitkerja' => 'required',
            'id_kdklasifikasi' => 'required',
            'id_kdtypoklasifikasi' => 'required',
            'TanggalQA' => 'required|before:today',
            'PenanggungJawab' => 'required',
            'LampiranDokumen' => 'required',
        ],
        [
            'NamaDokumen.required' => 'Nama Dokumen harus diisi',
            'id_kdunitkerja.required' => 'Harus dipilih',
            'id_kdklasifikasi.required' => 'Harus dipilih',
            'id_kdtypoklasifikasi.required' => 'Harus dipilih',
            'TanggalQA.required' => 'Harus dipilih',
            'TanggalQA.before' => 'Harus lebih kecil dari hari sekarang',
            'PenanggungJawab.required' => 'Harus dipilih',
            'LampiranDokumen.required' => 'Harus dilampirkan',
        ]
    );

    $dokumen = new Dokumen($request->input());

    $file = $request->lampiranDokumen;
    $filename = $file->getClientOriginalName();
    $path = public_path() . '/doc/';
    $file->move($path, $filename);
    $dokumen->lampiranDokumen = $filename;

    $tanggal = strtotime($request->TanggalQA);
    $newformat = date('Y-m-d', $tanggal);
    $dokumen->TanggalQA = $newformat;

    $dokumen->save();

    return redirect()->route('data.index');
}

```

Kode 4.8.3. Kode Data Controller - 3

```

        return redirect()->route('data.index');
    }

    /**
     * Display the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function show($id)
    {
        //
    }

    public function download($id)
    {
        $path = public_path() . '/doc/' . $id;
        return response()->download($path);
    }

    /**
     * Show the form for editing the specified resource.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function edit($id)
    {
        //
        $karyawan = Karyawan::all();
        $klasifikasi = Klasifikasi::all();
        $tipe = TipeKlasifikasi::all();
        $unit = UnitKerja::all();
        $dokumen = Dokumen::find($id);

        return view('edit', compact('karyawan', 'klasifikasi', 'tipe', 'unit', 'dokumen'));
    }

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request
     * @param int $id
     * @return \Illuminate\Http\Response
     */

```

Kode 4.8.4. Code Data Controller - 4

```

    /**
     * Update the specified resource in storage.
     *
     * @param \Illuminate\Http\Request $request, $id
     */
    public function update(Request $request, $id)
    {
        // return $id;
        $request->validate([
            [
                'namaDokumen' => 'required',
                'id_KdUnitKerja' => 'required',
                'id_KdKlasifikasi' => 'required',
                'id_KdTypeKlasifikasi' => 'required',
                'tanggalQA' => 'required|before:today',
                'penanggungjawab' => 'required',
            ],
            [
                'namaDokumen.required' => 'Nama Dokumen harus diisi',
                'id_KdUnitKerja.required' => 'Harus dipilih',
                'id_KdKlasifikasi.required' => 'Harus dipilih',
                'id_KdTypeKlasifikasi.required' => 'Harus dipilih',
                'tanggalQA.required' => 'Harus dipilih',
                'tanggalQA.before' => 'Harus lebih kecil dari hari sekarang',
                'penanggungjawab.required' => 'Harus dipilih',
            ]
        ]);

        $tanggal = strtotime($request->tanggalQA);
        $nowformat = date('Y-m-d', $tanggal);
        $dokumen = find($id)
            ->update([
                'namaDokumen' => $request->namaDokumen,
                'id_KdUnitKerja' => $request->id_KdUnitKerja,
                'id_KdKlasifikasi' => $request->id_KdKlasifikasi,
                'id_KdTypeKlasifikasi' => $request->id_KdTypeKlasifikasi,
                'tanggalQA' => $nowformat,
                'penanggungjawab' => $request->penanggungjawab
            ]);

        return redirect()->route('data.index');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */

```

Kode 4.8.5. Code Data Controller - 5


```

        return redirect()->route('data.index');
    }

    /**
     * Remove the specified resource from storage.
     *
     * @param int $id
     * @return \Illuminate\Http\Response
     */
    public function destroy($id)
    {
        $data = Dokumen::destroy($id);

        return redirect()->route('data.index');
    }
}

```

Kode 4.8.6. Kode Data Controller - 6

4.9. Model Dokumen

```

<?php
namespace App;

use DateTimeInterface;
use Illuminate\Database\Eloquent\Model;

class Dokumen extends Model
{
    protected $primaryKey = 'idDokumen';
    protected $table = 'dokumen';
    // protected $date = ['TanggalQA'];
    protected function serializeDate(DateTimeInterface $date)
    {
        return $date->format('d-m-Y H:i:s');
    }

    protected $fillable = ['NamaDokumen', 'id_KdUnitKerja', 'id_KdKlasifikasi', 'id_KdTypeKlasifikasi', 'TanggalQA', 'PenanggungJawab', 'LampiranDokumen'];

    public function unitkerja()
    {
        return $this->belongsTo('App\UnitKerja', 'id_KdUnitKerja', 'KdUnitKerja');
    }

    public function klasifikasi()
    {
        return $this->belongsTo('App\Klasifikasi', 'id_KdKlasifikasi', 'KdKlasifikasi');
    }

    public function typeklasifikasi()
    {
        return $this->belongsTo('App\TypeKlasifikasi', 'id_KdTypeKlasifikasi', 'KdTypeKlasifikasi');
    }

    public function karyawan()
    {
        return $this->belongsTo('App\Karyawan', 'PenanggungJawab', 'NIK');
    }
}

```

Kode 4.9. Kode Model Dokumen

4.10. Model Karyawan

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Karyawan extends Model
{
    protected $table = "karyawan";

    public function dokumen()
    {
        return $this->hasMany('App\Dokumen');
    }
}
```

Kode 4.10. Kode Model Karyawan

4.11. Model Klasifikasi

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Klasifikasi extends Model
{
    protected $table = "klasifikasi";

    public function dokumen()
    {
        return $this->hasMany('App\Dokumen');
    }
}
```

Kode 4.11. Kode Model Klasifikasi

4.12. Model TipeKlasifikasi

```
<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class TipeKlasifikasi extends Model
{
    protected $table = "typeklasifikasi";

    public function dokumen()
    {
        return $this->hasMany('App\Dokumen');
    }
}
```

Kode 4.12. Kode Model TipeKlasifikasi

4.13. Model UnitKerja

```
<?php

namespace App;


use Illuminate\Database\Eloquent\Model;

class UnitKerja extends Model
{
    protected $table = "unitkerja";

    public function dokumen()
    {
        return $this->hasMany('App\Dokumen');
    }
}
```

Kode 4.13. Kode Model Unit Kerja

4.14. Model User

```
php

namespace App;

use Illuminate\Notifications\Notifiable;
use Illuminate\Contracts\Auth\MustVerifyEmail;
use Illuminate\Foundation\Auth\User as Authenticatable;
use Spatie\Permission\Traits\HasRoles;

class User extends Authenticatable
{
    use Notifiable, HasRoles;

    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password',
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];

    /**
     * The attributes that should be cast to native types.
     *
     * @var array
     */
    protected $casts = [
        'email_verified_at' => 'datetime',
    ];
}
```

Kode 4.14. Kode Model User

4.15. Migrasi Tabel User

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->timestamp('email_verified_at')->nullable();
            $table->string('password');
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('users');
    }
}
```

Kode 4.15. Kode Migrasi Tabel User

4.16. Migrasi Tabel User Permission

```
//php
use Illuminate\Support\Facades\Schema;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreatePermissionTables extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        $tableNames = config('permission.table_names');
        $columnNames = config('permission.column_names');

        if (empty($tableNames)) {
            throw new \Exception('Error: config/permission.php not loaded. Run [php artisan config:clear] and try again.');
```

```
        }

        Schema::create($tableNames['permissions'], function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('guard_name');
            $table->timestamps();
        });

        Schema::create($tableNames['roles'], function (Blueprint $table) {
            $table->bigIncrements('id');
            $table->string('name');
            $table->string('guard_name');
            $table->timestamps();
        });

        Schema::create($tableNames['model_has_permissions'], function (Blueprint $table) use ($tableNames, $columnNames) {
            $table->unsignedBigInteger('permission_id');

            $table->string('model_type');
            $table->unsignedBigInteger($columnNames['model_morph_key']);
            $table->index([$columnNames['model_morph_key'], 'model_type'], 'model_has_permissions_model_id_model_type_index');

            $table->foreign('permission_id')
                ->references('id')
                ->on($tableNames['permissions'])
```

Kode 4.16.1. Kode Migrasi Tabel User Permission

```
$table->foreign('permission_id')
    ->references('id')
    ->on($tableNames['permissions'])
    ->onDelete('cascade');

$table->primary(['permission_id', $columnNames['model_morph_key'], 'model_type'],
    'model_has_permissions_permission_model_type_primary');
});

Schema::create($tableNames['model_has_roles'], function (Blueprint $table) use ($tableNames, $columnNames) {
    $table->unsignedBigInteger('role_id');

    $table->string('model_type');
    $table->unsignedBigInteger($columnNames['model_morph_key']);
    $table->index([$columnNames['model_morph_key'], 'model_type'], 'model_has_roles_model_id_model_type_index');

    $table->foreign('role_id')
        ->references('id')
        ->on($tableNames['roles'])
        ->onDelete('cascade');

    $table->primary(['role_id', $columnNames['model_morph_key'], 'model_type'],
        'model_has_roles_role_model_type_primary');
});

Schema::create($tableNames['role_has_permissions'], function (Blueprint $table) use ($tableNames) {
    $table->unsignedBigInteger('permission_id');
    $table->unsignedBigInteger('role_id');

    $table->foreign('permission_id')
        ->references('id')
        ->on($tableNames['permissions'])
        ->onDelete('cascade');

    $table->foreign('role_id')
        ->references('id')
        ->on($tableNames['roles'])
        ->onDelete('cascade');

    $table->primary(['permission_id', 'role_id'], 'role_has_permissions_permission_id_role_id_primary');
});

app('cache')
    ->store(config('permission.cache.store') != 'default' ? config('permission.cache.store') : null)
    ->forget(config('permission.cache.key'));
```

Kode 4.16.2. Kode Migrasi Tabel User Permission - 2


```

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    $tableNames = config('permission.table_names');

    if (empty($tableNames)) {
        throw new \Exception('Error: config/permission.php not found and defaults could not be merged. Please publish the package configuration before proceeding, or drop the table manually.');
    }

    Schema::drop($tableNames['role_has_permissions']);
    Schema::drop($tableNames['model_has_role']);
    Schema::drop($tableNames['model_has_permissions']);
    Schema::drop($tableNames['roles']);
    Schema::drop($tableNames['permissions']);
}
}

```

Kode 4.16.3. Kode Migrasi Tabel User Permission - 3

4.17. Migrasi Tabel Unit Kerja

```

<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateUnitkerjaTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('unitkerja', function (Blueprint $table) {
            $table->char('KdUnitKerja',3)->primary();
            $table->char('NmUnitKerja',50);
            $table->char('Inputer',20);
            $table->timestamps();
            // $table->primary('KdUnitKerja');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('unitkerja');
    }
}

```

Kode 4.17. Kode Migrasi Tabel Unit Kerja

4.18. Migrasi Tabel Klasifikasi

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateKlasifikasiTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('klasifikasi', function (Blueprint $table) {
            $table->char('KdKlasifikasi',2)->primary();
            $table->char('NmKlasifikasi',30);
            $table->char('Inputer',20);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('klasifikasi');
    }
}
```

Kode 4.18. Kode Migrasi Tabel Klasifikasi

4.19. Migrasi Tabel Tipe Klasifikasi

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateTypeKlasifikasiTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('typeklasifikasi', function (Blueprint $table) {
            $table->char('KdTypeKlasifikasi', 4)->primary();
            $table->char('NmTipeKlasifikasi',30);
            $table->char('Inputer',20);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('typeklasifikasi');
    }
}
```

Kode 4.19. Kode Migrasi Tipe Klasifikasi

4.20. Migrasi Tabel Karyawan

```
<?php

use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateKaryawanTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('karyawan', function (Blueprint $table) {
            $table->char('NIK',12)->primary();
            $table->char('NmKaryawan',50);
            $table->char('Inputer',20);
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('karyawan');
    }
}
```

Kode 4.20. Kode Migrasi Tabel Karyawan

4.21. Migrasi Tabel Dokumen

```
<?php
use Illuminate\Database\Migrations\Migration;
use Illuminate\Database\Schema\Blueprint;
use Illuminate\Support\Facades\Schema;

class CreateDokumenTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('dokumen', function (Blueprint $table) {
            $table->id('IdDokumen');

            $table->char('Id_KdUnitKerja', 3);
            $table->foreign('Id_KdUnitKerja', 3)->references('KdUnitKerja')->on('unitkerja')->onDelete('cascade');

            $table->char('Id_KdKlasifikasi', 2);
            $table->foreign('Id_KdKlasifikasi', 2)->references('KdKlasifikasi')->on('klasifikasi')->onDelete('cascade');

            $table->char('PenanggungJawab', 12);
            $table->foreign('PenanggungJawab', 12)->references('NIK')->on('karyawan')->onDelete('cascade');

            $table->char('Id_KdTypeKlasifikasi', 4);
            $table->foreign('Id_KdTypeKlasifikasi', 4)->references('KdTypeKlasifikasi')->on('typeklasifikasi')->onDelete('cascade');

            $table->char('NamaDokumen', 60);

            $table->date('TanggalQA')->nullable();
            $table->date('TanggalRekam');
            $table->binary('LampiranDokumen');
            $table->timestamps();
        });

        // Schema::table('dokumen', function (Blueprint $table) {
            // });
    }

    /**
     * Reverse the migrations.
     *
     */
}
```

Kode 4.21.1. Kode Migrasi Tabel Dokumen

```
        // Reverse the migrations.
        *
        * @return void
        */
        public function down()
        {
            Schema::dropIfExists('dokumen');
        }
    }
}
```

Kode 4.21.2. Kode Fungsi Drop Tabel Dokumen jika exist

[Halaman ini sengaja dikosongkan]

BAB V

PENGUJIAN DAN EVALUASI

Bab ini menjelaskan tahap uji coba dilakukan terhadap perangkat lunak.

5.1. Tujuan Pengujian

Pengujian dilakukan terhadap aplikasi perekaman dokumen untuk menguji kesesuaian dan ketepatan fungsionalitas dari seluruh sistem aplikasi dengan menguji tingkat guna aplikasi.

5.2. Kriteria Pengujian

Penilaian atas pencapaian tujuan pengujian aplikasi didapatkan dengan memperhatikan kesesuaian output dengan tujuan fungsi tersebut.

5.3. Skenario Pengujian

Skenario pengujian dilakukan dengan sebuah user sebagai admin dengan menambah rekaman dokumen, menghapus rekaman dokumen, mengubah rekaman dokumen, mengupload file dokumen dan memfilter rekaman dokumen.

Terdapat beberapa skenario pengujian, yaitu:

1. Admin melakukan login dengan mengisi email dan password yang sudah terdaftar.
2. Admin menambah rekaman dokumen dengan membuka halaman rekam dokumen yang ada disebelah kiri dan mengisi data-data seputar dokumen yang mau disimpan.

3. Admin menginput file dokumen kedalam perekaman dan menekan tombol simpan.

```
lluminate\Http\Request {#44 ▼
  #json: null
  #convertedFiles: null
  #userResolver: Closure($guard = null) {#1196 ▶}
  #routeResolver: Closure() {#1205 ▶}
  +attributes: Symfony\Component\HttpFoundation\ParameterBag {#46 ▶}
  +request: Symfony\Component\HttpFoundation\ParameterBag {#45 ▼
    #parameters: array:7 [▼
      "_token" => "QsE11LH28Svh46VE8Lft2N18XgKA11SWvSow8Zg"
      "NamaDokumen" => "Pembelian"
      "id_KdUnitKerja" => "002"
      "id_KdKlasifikasi" => "k1"
      "id_KdTypeKlasifikasi" => "tk1"
      "TanggalQA" => "18-12-2020"
      "PenanggungJawab" => "k001"
    ]
  }
  +query: Symfony\Component\HttpFoundation\InputBag {#52 ▶}
  +server: Symfony\Component\HttpFoundation\ServerBag {#49 ▶}
  +files: Symfony\Component\HttpFoundation\FileBag {#48 ▶}
  +cookies: Symfony\Component\HttpFoundation\InputBag {#47 ▶}
  +headers: Symfony\Component\HttpFoundation\HeaderBag {#50 ▶}
  #content: null
  #languages: null
  #charset: null
  #encodings: null
  #acceptableContentTypes: null
  #pathInfo: "/data"
  #requestUri: "/data"
  #baseUrl: ""
  #basePath: null
  #method: "POST"
  #format: null
  #session: Illuminate\Session {#1238 ▶}
  #locale: null
  #defaultLocale: "en"
  -preferredFormat: null
  -isHostValid: true
  -isForwardedValid: true
  -isSafeContentPreferred: null
  basePath: ""
  format: "html"
}
```

Gambar 5.3.1. Pengujian Input Dokumen

4. Dokumen yang disimpan dapat dilihat pada halaman pencarian dokumen.

5. Admin memfilter dokumen berdasarkan nama dokumen atau tanggal periode dokumen dibuat dan diupload.

```
Illuminate\Http\Request {#43 ▾  
  #json: null  
  #convertedFiles: []  
  #userResolver: Closure($guard = null) {#1202 ▸}  
  #routeResolver: Closure() {#1203 ▸}  
  #attributes: Symfony\Component\HttpFoundation\ParameterBag {#45 ▸}  
  #request: Symfony\Component\HttpFoundation\InputBag {#51 ▸}  
  #query: Symfony\Component\HttpFoundation\InputBag {#51 ▸  
    #parameters: array:2 {  
      "_token" => "sxHg655HaYBpIXOHCAoVhPqIu7KejC1wPh2UDtK"  
      "car1" => "Perencanaan"  
    }  
  }  
  #server: Symfony\Component\HttpFoundation\ServerBag {#47 ▸}  
  #files: Symfony\Component\HttpFoundation\FileBag {#48 ▸}  
  #cookies: Symfony\Component\HttpFoundation\CookieBag {#46 ▸}  
  #headers: Symfony\Component\HttpFoundation\HeaderBag {#49 ▸}  
  #content: null  
  #languages: null  
  #charset: null  
  #encoding: null  
  #acceptableContentTypes: null  
  #pathInfo: "/search"  
  #requestUri: "/search?_token=sxHg655HaYBpIXOHCAoVhPqIu7KejC1wPh2UDtK&car1=Perencanaan"  
  #baseUri: ""  
  #basePath: null  
  #method: "GET"  
  #format: null  
  #session: Illuminate\Session {#1241 ▸}  
  #locale: null  
  #defaultLocale: "en"  
  #preferredFormat: null  
  -isHostValid: true  
  -isForwardedValid: true  
  -isSafeContentPreferred: null  
  #basePath: ""  
  #format: "html"  
}
```

Gambar 5.3.2. Pengujian Filter Dokumen dengan Nama Dokumen

```

Illuminate\Http\Request {#43 ▼
  #json: null
  #convertedFiles: []
  #userResolver: Closure($guard => null) {#1202 ▶}
  #routeResolver: Closure() {#1203 ▶}
  #attributes: Symfony\Component\HttpFoundation\ParameterBag {#45 ▼
    #parameters: []
  }
  #request: Symfony\Component\HttpFoundation\InputBag {#51 ▶}
  #query: Symfony\Component\HttpFoundation\InputBag {#51 ▼
    #parameters: array:3 ▼
      "_token" => "a0qg55HAYbpIXOHCAavVhPqM7KqjC1MFPh2UDsk"
      "mulai" => "01-08-2020"
      "akhir" => "31-08-2020"
    }
  }
  #server: Symfony\Component\HttpFoundation\ServerBag {#47 ▼
    #parameters: array:29 [▶]
  }
  #files: Symfony\Component\HttpFoundation\FileBag {#48 ▼
    #parameters: []
  }
  #cookies: Symfony\Component\HttpFoundation\CookieBag {#46 ▼
    #parameters: array:2 [▶]
  }
  #headers: Symfony\Component\HttpFoundation\HeaderBag {#49 ▼
    #headers: array:13 [▶]
      "host" => array:1 [▶]
      "connection" => array:1 [▶]
      "upgrade-insecure-requests" => array:1 [▶]
      "user-agent" => array:1 [▶]
      "accept" => array:1 [▶]
      "sec-fetch-site" => array:1 [▶]
      "sec-fetch-mode" => array:1 [▶]
      "sec-fetch-user" => array:1 [▶]
      "sec-fetch-dest" => array:1 [▶]
      "referer" => array:1 [▶]
      "accept-encoding" => array:1 [▶]
      "accept-language" => array:1 [▶]
      "cookie" => array:1 [▶]
    }
    #cacheControl: []
  }
  #content: null
  #languages: null
  #charsets: null
  #encodings: null
  #acceptableContentTypes: null
}

```

Gambar 5.3.3. Pengujian Filter Dokumen dengan Tanggal Dokumen

- Admin dapat menghapus rekaman dokumen dengan menekan tombol hapus yang ada pada halaman pencarian dokumen, aplikasi akan mencari id dokumen dan menghapus dokumen.

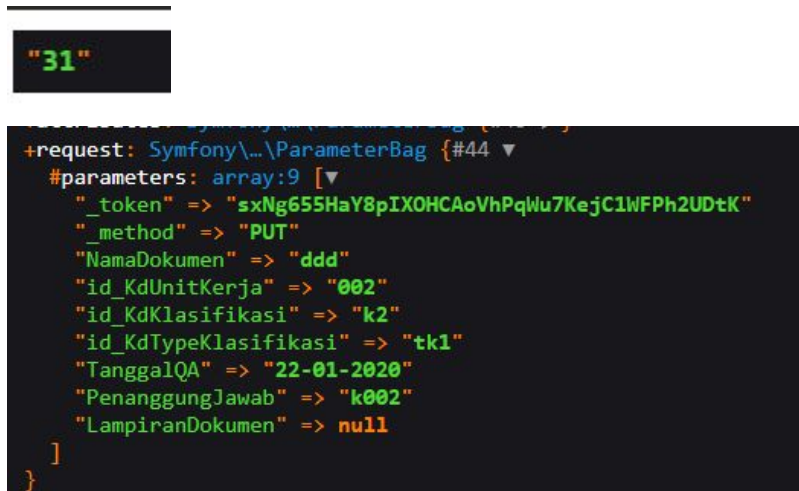
1	kmlkml	DIVISI KEPESERTAAN	Jaringan	Budi	Change Requirement	19-08- 2020	29-08- 2020	Database ABISS 2019- 2020.xlsx	<div>Ubah</div> <div>Hapus</div>
---	--------	-----------------------	----------	------	-----------------------	----------------	----------------	---	----------------------------------

Gambar 5.3.3. Pengujian Menghapus Dokumen melalui ID



- Admin dapat mengedit dokumen dengan menekan tombol edit yang ada di halaman pencarian dokumen

dan mengisi kembali data dokumen yang mau diubah lalu disimpan.



Gambar 5.3.4. Pengujian MengeditDokumen melalui ID

5.4. Evaluasi Pengujian

Pembimbing lapangan melakukan pengujian terhadap aplikasi perekaman dokumen. Berikut ini adalah hasil uji coba aplikasi yang telah dibuat :

Kebutuhan	Deskripsi Kebutuhan
Menambah rekaman dokumen	Terpenuhi
Mengubah rekaman dokumen	Terpenuhi
Menghapus rekaman dokumen	Terpenuhi
Mengupload file	Terpenuhi

Filter dokumen	Terpenuhi
----------------	-----------

Tabel 5.4.. Tabel Evaluasi Pengujian Kebutuhan

Dengan hasil pengujian pada tabel di atas, dapat disimpulkan bahwa secara keseluruhan fungsionalitas terpenuhi.

BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Kesimpulan yang didapat setelah melakukan pembuatan dashboard pada kegiatan kerja praktik di PT. TASPEN(Persero) adalah sebagai berikut:

- Dengan adanya aplikasi web ini, para karyawan menjadi mudah untuk melihat dokumen yang telah tersimpan. Karyawan juga dapat menyimpan dokumen dan juga melakukan edit pada detail dokumen.
- Membangun aplikasi web dengan framework Laravel dirasa sangat cocok dikarenakan Laravel adalah framework yang banyak dipakai oleh khalayak ramai.

6.2. Saran

Saran untuk pengembangan sistem aplikasi perekaman dokumen adalah sebagai berikut:

- Dilakukan pembaruan data pada database setiap ada perubahan karyawan. Agar data bisa selalu ter update
- Optimasi UI yang lebih baik

DAFTAR PUSTAKA

- [1] Laravel, “Laravel”, [Online]
<https://laravel.com/docs/4.2/introduction>
- [2] PHP, “PHP”, [Online]
<https://www.tutorialspoint.com/php/index.htm>
- [3] Bootstrap, “Bootstrap”, [Online]
<https://getbootstrap.com/>
- [4] JavaScript, “JavaScript”, [Online]
<https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- [5] Spatie, “Spatie”, [Online]
<https://spatie.be/docs/laravel-permission/v3/introduction>
- [6] MySQL, “MySQL”, [Online]
<https://www.mysql.com/>

BIODATA PENULIS I

Nama : Raja Permata Boy Mangatur Simarmata
Tempat, Tanggal Lahir : Pematang Siantar, 23 Mei 1999
Jenis Kelamin : Laki-laki
Agama : Kristen
Status : Belum Menikah
Alamat Asal : Jln Gang Setia No. 6, Pematang Siantar,
Sumatera Utara
Alamat Surabaya : Jln Keputih Perintis 4 No 7A, Sukolilo,
Surabaya
Telepon : 082239366960
Email : raja.simarmata23@gmail.com

PENDIDIKAN FORMAL

2017 – sekarang : S1 Informatika ITS
2014 – 2017 : SMA DEL Laguboti
2011 – 2014 : SMP St. Fransiskus Assisi Samarinda
2005 – 2011 : SD St. Maria Pekanbaru

KEMAMPUAN

- *Web Programming* (HTML, PHP, CSS, Javascript)
- *Programming* (C, C++, Python)
- *Database Manajemen* (MySQL)
- *Software Perkantoran* (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Informatika, Fakultas Teknologi
Informasi dan Komunikasi, Institut Teknologi
Sepuluh Nopember Surabaya
Angkatan : 2017
Semester : 7 (Tujuh)
IPK : 3.35

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS II

Nama : Edgar Andrew Hutaaruk
Tempat, Tanggal Lahir : Pematangsiantar, 28 Juni 1999
Jenis Kelamin : Laki-laki
Agama : Katolik
Status : Belum menikah
Alamat Asal : Jl. Pisang Kipas, Kota Pematangsiantar
Alamat Surabaya : Keputih perintis IV no.7b, Sukolilo, Kota Surabaya
Telepon : 082160313277
Email : edgarandrew699@gmail.com

PENDIDIKAN FORMAL

2017 – sekarang : S1 Informatika ITS
2014 – 2017 : SMA Budi Mulia Pematangsiantar
2012 – 2014 : SMP Methodist Pematangsiantar
2011 – 2012 : SMP Gembala Baik Pontianak
2009 – 2011 : SD Gembala Baik Pontianak
2005 – 2009 : SD Methodist Pematangsiantar

KEMAMPUAN

- *Web Programming* (Laravel, HTML, PHP, CSS, Javascript)
- *Programming* (C, C++, Python)
- *Database Management* (MySQL)
- *Software* Perkantoran (Microsoft Word, Excel, PowerPoint)
- Bahasa (Indonesia, Inggris)

AKADEMIS

Kuliah : Departemen Informatika – Fakultas Teknologi Informasi dan Komunikasi, Institut Teknologi Sepuluh Nopember Surabaya
Angkatan : 2017
Semester : 7 (Tujuh)
IPK : 3.04

[Halaman ini sengaja dikosongkan]